

Grammar\_Elem:

Kind:

- Nonterminal
  - Terminal
- > the kind of the elem

std::string str

-> the string representation

Grammar\_Production:

Grammar\_Elem nonterminal;

-> the nonterminal at the left hand side of the production

std::vector<Grammar\_Elem> elems;

-> the list of elems at the right hand side of the production

Grammar:

Grammar\_Elem starting\_nonterminal;

-> starting nonterminal

std::vector<Grammar\_Elem> elems;

-> the elems

std::vector<Grammar\_Production> productions;

-> the productions

Grammar Grammar\_read\_from\_file(std::string path)

-> reads the grammar from a file with the syntax of

grammar = starting\_nonterminal productions.

starting\_nonterminal = "starting\_nonterminal:" nonterminal ";".

productions = "productions:" {production} ";".

production = nonterminal ":" {(nonterminal | terminal) "," } ";".

nonterminal = (letter | digit) { (letter | digit) }.

digit = "0" | ... | "9".

letter = "a" | ... | "z" |

"A" | ... | "Z".

terminal = "\"" (\* any character except "\" ) "\".

bool Grammar\_cfg\_check(Grammar grammar)

-> checks if the grammar is cfg